SKILLWEAVER: FULL ARCHITECTURE, ROADMAP & FEATURE SET

----

1. ARCHITECTURE OVERVIEW

SkillWeaver is a modular macro engine built for World of Warcraft, replacing GSE with a deterministic, scalable, and maintainable system.

A) Core Architecture Layers

1. Engine Layer

- SequenceEngine.lua: deterministic macro execution
- ConditionParser.lua: parses conditional logic
- ResourceCheck.lua: evaluates class resources
- CooldownLogic.lua: manages prioritization

2. Database Layer

- SkillWeaverDB.lua: all class/spec builds
- TalentDB.lua: talent import strings (Wowhead)
- PvPTalentDB.lua: PvP talents
- HeroTalentDB.lua: hero trees

3. UI Layer

- ModeSelector.lua
- Editor.lua
- Browser.lua
- Options.lua
- XML frames

4. Sync Layer (External)

- Web scraper
- Sync client
- SavedVariables updater
- Docker deployment

B) Addon Folder Structure

```
~/Documents/skillweaver/
core/
engine/
db/
talents/
pvp/
heroes/
sequences/
ui/
utils/
libs/
SkillWeaver.toc
```

----

2. FEATURE SET

A) Macro System

- Deterministic sequencing
- Multi-mode support (Mythic+, Raid, PvP, Delves)
- Balanced + High Performance engines
- Smart AoE detection
- Cooldown alignment logic
- Proc-aware execution
- Conditional resource evaluation
- Defensive logic integration

B) Talent System

- Full Wowhead import support
- PvP talents per spec
- Hero talent tree integration
- Build versioning
- Mismatch detection

- Auto-suggest talents based on mode

C) Update System

- Server-side web scraping (Wowhead, Icy-Veins, Method)
- Automatic weekly build refresh
- Desktop sync client
- Build metadata + versioning
- Cross-platform (Windows/Mac/Linux)
- Docker-based scraper

D) UI Features

- Talent viewer/editor
- Sequence browser
- Mode selection (Raid/M+/PvP/Delve)
- Sequence editor with syntax validation
- Warnings for unsupported talents
- Copy-to-clipboard Wowhead code
- Import/export panels

E) Developer Features

- Modular sequence files
- JSON-based scraper output
- API hooks for community modules
- Class-level hot-reload
- Debug mode

----

3. DEVELOPMENT ROADMAP

PHASE 1: CORE ENGINE [IN PROGRESS]

- Create TOC file
- Build core namespace
- SequenceEngine implementation

- ConditionParser + evaluator
- ResourceCheck + CooldownLogic
- Global initialization routines

PHASE 2: UI SYSTEM

- Frame XMLs
- Mode selector UI
- Talent import UI
- Sequence editor
- Browser/search/filter interface

PHASE 3: DATABASE SKELETON

- Create SkillWeaverDB root
- Class/spec directories set up
- Talent/Hero/PvP DB foundations
- Mode stubs for each spec

PHASE 4: CLASS MODULE GENERATION (12 PASSES)

For each class:

- Mythic+ builds (Balanced + HighPerf)
- Raid ST/Cleave builds
- PvP builds with PvP talents
- Open World / Delve builds
- Sequence generation
- Wowhead imports
- Hero talents
- Metadata
- DB merge

PHASE 5: SYNC CLIENT

- Python OR Node.js client
- Docker scraper container
- Scheduler (cron)
- Build normalization

- SavedVariables updater

PHASE 6: FINALIZATION

- Merge full DB
- Generate SavedVariables template
- Validate all sequences
- Optional WeakAura companion buttons
- Packaging + documentation

----

4. PROJECT GOALS

- Become the successor to GSE
- Multi-mode macro engine with deterministic logic
- Fully automated build/talent updates
- Per-spec Balanced and High Performance rotations
- PvP-aware sequences with PvP talents
- Full external toolchain (scraper + sync client)
- Long-term extensibility

END OF DOCUMENT
